# COMPOUND KEY WORD GENERATION FROM DOCUMENT DATABASES USING A HIERARCHICAL CLUSTERING ART MODEL

Aberto Muñoz

96-76

WORKING PAPERS

Universidad Carlos III de Madrid

Working Paper 96-76
Statistics and Econometrics Series 31
January 1997

Departamento de Estadística y Econometría
Universidad Carlos III de Madrid
Calle Madrid, 126
28903 Getafe (Spain)
Fax (341) 624-9849

# COMPOUND KEY WORD GENERATION FROM DOCUMENT DATABASES USING A HIERARCHICAL CLUSTERING ART MODEL.

Alberto Muñoz*

Abstract

The growing availability of databases on the information highways motivates the development of new processing tools able to deal with a heterogeneous and changing information environment. A highly desirable feature of data processing systems handling this type of information is the ability to automatically extract its own key words. In this paper we address the specific problem of creating semantic term associations from a text database. The proposed method uses a hierarchical model made up of Fuzzy Adaptive Resonance Theory (ART) neural networks. First, the system uses several Fuzzy ART modules to cluster isolated words into semantic classes, starting from the database raw text. Next, this knowledge is used together with coocurrence information to extract semantically meaningful term associations. These associations are asymmetric and one-to-many due to the polisemy phenomenon. The strength of the associations between words can be measured numerically. Besides this, they implicitly define a hierarchy between descriptors. The underlying algorithm is appropriate for employment on large databases. The operation of the system is illustrated on several real databases.

Keywords:
Automatic indexing, knowledge extraction, information retrieval, neural Fuzzy ART models.

*Muñoz, Department of Statistics and Econometrics, Universidad Carlos III de Madrid; The author wishes to thank the support from the DGICYT PB94-0374 Spanish grant. The author is also grateful to two anonymous referees for their valuable comments and remarks.

# Compound key word generation from document databases using a hierarchical clustering ART model

Alberto Muñoz

Department of Statistics and Econometrics
University Carlos III, 28903 Getafe, Madrid, España
e-mail: albmun@est-econ.uc3m.es

## Abstract

The growing availability of databases on the information highways motivates the development of new processing tools able to deal with a heterogeneous and changing information environment. A highly desirable feature of data processing systems handling this type of information is the ability to automatically extract its own key words. In this paper we address the specific problem of creating semantic term associations from a text database. The proposed method uses a hierarchical model made up of Fuzzy Adaptive Resonance Theory (ART) neural networks. First, the system uses several Fuzzy ART modules to cluster isolated words into semantic classes, starting from the database raw text. Next, this knowledge is used together with coocurrence information to extract semantically meaningful term associations. These associations are asymmetric and one-to-many due to the polisemy phenomenon. The strength of the associations between words can be measured numerically. Besides this, they implicitly define a hierarchy between descriptors. The underlying algorithm is appropriate for employment on large databases. The operation of the system is illustrated on several real databases.

**Key words:** Automatic indexing, knowledge extraction, information retrieval, Neural Fuzzy ART models, information retrieval

## 1 Introduction

A well known problem in Information Retrieval (IR) research is the difficulty faced by users who try to accurately formulate queries to retrieve the documents they want. This problem has to do with the user's lack of familiarity with the database lexicon, and often results in a not very satisfactory retrieved document set [20,49]. In order to overcome this problem, many commercial databases include a thesaurus which helps users to find the precise terms that match documents they are seeking. Throughout this work, expressions *word*, *term*, *descriptor* and *key word* will occasionally be interchanged. Since the information stored in many databases (as in the case of networked information in Internet) changes continuously, it is not generally feasible to update a thesaurus manually; often, the absence of a uniform structure in databases makes this task even harder. In the case of networked information, the possibility of browsing documents makes the thesaurus an essential part of the system. If no thesaurus is used, the problem of *documental noise* arises. To illustrate this problem, let us show an example. We formulated a query with the term "*cluster*" using the Lycos program [17] in Internet. The system returned a set of 9390 documents. But in the sense we wanted, "*cluster analysis*", there were only 91 documents, a smaller set for visual inspection. The use of compound key words in queries, more specific than single descriptors, helps to simplify these situations [46].

A typical thesaurus provides for every term in it a list of broader, narrower and related terms. Thesauri can be manual or automatic. Automatic thesauri are dependent on the corpus used in their elaboration, since they use term coocurrence in documents to build the associations [16,46,26]. On the other hand, this feature allows them to keep up-to-date in a changing information environment.

Concerning previous work in compound key word generation, Salton [46] proposes a procedure based on word coocurrence inside sentences. To form a key word, a main term is chosen (imposing a frequency threshold) and then associations are made between this term and other terms in the same sentence. The system is strongly arbitrary concerning the choice of the main term, and can produce a high percentage of meaningless associations. In [26] some new methods for thesaurus generation and their performance in IR tasks are reviewed. In that paper the authors point out some problems of these systems, as the need to use relevance judgements to generate key words. They also comment on the difficulty in evaluating the performance of thesauri.

In this paper we focus on the automatic generation of compound key words using frequency of words in documents. The process starts with a list of single index terms from the database under study. This list is automatically generated from the database raw text using standard, simple indexing techniques, as explained in subsection 2.1. For the task of generating key words, single terms are first clustered into groups of semantically related words. We will call these groups "*semantic classes*" in the sequel. Given a particular key word, the words that form it are semantically related and, therefore, they are expected to belong to the same semantic classes. Thus, given a single term, the search of co-ocurrent terms in key words for that term can be restricted to the semantic classes to which that word belongs. In this way, the search space for semantic associations is drastically reduced, and the risk of generating meaningless associations is strongly reduced too.

The rest of the paper is as follows: the structure of document information is described in section 2 as well as some particularities for this type of information. In section 3 a neural network based clustering architecture is introduced for the task of semantic class formation. Compound key words are generated using information from these semantic classes. Section 4 is devoted to experimental work on two real databases. Finally, section 5 summarizes and outlines some new research tasks.

## 2 Structure of document information

### 2.1 The vector space model

In this subsection we briefly describe the "*vector space model*" [45,46], used in this paper to represent documents and terms. This model allows a symmetric treatment between terms and documents. We will see that, despite the duality existing between term and document vectors, there are differences that will prevent us from using the same clustering algorithms on terms that are often used on documents.

In the vector space model both documents and terms are represented as points of a vector space:

Let $t_1, t_2, \dots, t_n$ denote the terms used for indexing the database and $D_1, D_2, \dots, D_m$ the documents in the database. Document $D_i$ is represented by:

$$D_i = (a_{i1}, a_{i2}, \dots, a_{in}) \tag{1}$$

where $a_{ij}$ is the weight of term $t_j$ in document $D_i$.

Term $t_j$ is represented by:

$$t_j = (a_{1j}, a_{2j}, \dots, a_{mj}) \tag{2}$$

In this way the *term-document matrix* M results:

$$\begin{array}{lcccc}
\mathrm{M}\ - & t_1 & t_2 & \dots & t_n \\
\hline
D_1 - & a_{11} & a_{12} & \dots & a_{1n} \\
D_2 - & a_{21} & a_{22} & \dots & a_{2n} \\
\cdot\ \cdot\ \cdot & \cdot & \cdot & & \\
D_m - & a_{m1} & a_{m2} & \dots & a_{mn}
\end{array} \tag{3}$$

If coefficients $a_{ij}$ are chosen so that $a_{ij} = 1$ when term $t_j$ is present in document $D_i$, $a_{ij} = 0$ in other case, binary vectors are obtained. To get real-valued vectors, the TF-IDF (Term Frequency - Inverse Document Frequency) method [47] is broadly used. Let us consider the following quantities:

$n_j$ = number of documents indexed by term $t_j$
$m$ = number of documents in database
$t_{ij}$ = number of occurrences of term $t_j$ in document $D_i$

The IDF method chooses:

$$a_{ij} = t_{ij} \cdot \log(\frac{m}{n_j}) \tag{4}$$

The IDF method has been recently justified using an information theoretic approach [50]. For our purposes only presence/absence information of terms in documents is needed, so we will adopt the binary scheme in the sequel. Queries formulated to the database are represented like documents in the vector space model.

When a query is formulated, a "*similarity measure*" is used for ranking documents according to relevance to the query. Some commonly used similarity measures are the inner product and the cosine measure for real-valued vectors, and the Dice coefficient or the Jaccard coefficient for binary-valued vectors. For a more comprehensive list, see [46].

An open question is how to select the single index terms $t_1, t_2, \dots, t_n$. The most commonly used method for this task [11,46] works as follows: first of all, an stop-word list is used to remove common words from the database, such as "the", "that", "of", etc. Next, both rare words and too frequent words are removed using cut thresholds in the term frequency distribution. In this way, words with maximum semantic discrimination power are expected to remain.To date, there is no general rule to choose these frequency thresholds.

Here, this method for index term selection will be used, but introducing a modification. The method just described removes very frequent words from the vocabulary. Assuming empty words have been removed, frequent words are often present in key words. If we remove frequent words such as "information", "higher" or "needs", important key words such as "information retrieval", "higher education" or "user needs" will be lost. This is why we will not impose an upper frequency threshold to exclude frequent terms from databases. It is appropriate to say here as well that single words are not previously stemmed, and that the empty-word list used remains unchanged in all experiments.

### 2.2 Asymmetry between rows and columns in the term-document matrix

Considering equations 1 and 2, it is apparent the duality existing between term and document vectors: documents are the rows of M (eq. 3) and terms are the columns. Suppose now we want to perform a cluster analysis on a given document set. Let us denote $s_{ij}$ the similarity between document i-th and document j-th. All the information needed to carry out a clustering process is contained in the similarity matrix S=($s_{ij}$). For example, Jaccard's coefficient is computed by:

$$s_{ij} = \frac{|D_i \wedge D_j|}{|D_i| + |D_j| - |D_i \wedge D_j|} \quad , \quad s_{ij}^{*} = \frac{\sum_{k=1}^{n} a_{ik} a_{jk}}{\sum_{k=1}^{n} a_{ik}^2 + \sum_{k=1}^{n} a_{jk}^2 - \sum_{k=1}^{n} a_{ik} a_{jk}} \qquad (5)$$

where $|D_i|$ = number of $a_{ij} \neq 0$, i.e., number of present terms in $D_i$, and $|D_i \wedge D_j|$ = number of common terms between $D_i$ and $D_j$. $s_{ij}$ is used for binary vectors and $s_{ij}^{*}$ for real-valued vectors.

We see that computation of Jaccard's coefficient only requires vector representations for each $D_i \in \Re^n$. This is true for all commonly used similarity measures [1,19,25]. For this reason the same algorithms can equally be used to cluster documents and terms. Some references on clustering documents are [7,22,28].

A word of caution on the duality between terms and documents is necessary however. It is well known that word frequencies in documents roughly follow a Zipf's law [52]: If terms are arranged by descending occurrence order, where occurence is the number of documents containing the given word, the frequency of term in position r (r=1,2,3,...), f(r) say, verifies

$$f(r)\, r \approx k \text{ , where k is a constant.} \qquad (6)$$

More accurate forms of this law can be found in [38,39]. From Zipf's law it follows that in any database there will be many rare terms but just a few very frequent terms. A concrete example is shown in figure 1. The feature used to produce the histogram is the number of documents in which terms occur. The histogram is made from the vocabulary (near 6000 different words) of the database used in example 4.1, which contains more than 5000 documents covering the common topic `information retrieval'. The apparent positive skewness indicates that most words occur in few documents, i.e., they are rare words. It is readily seen that there are very few words occurring more than in 100 documents (despite the fact that all documents deal with the same main topic).

Binary representations of rare terms will have many 0's and very few 1's; the inverse situation will happen for very frequent terms. Consider now a concrete example. Let A=`statistics' and B=`regression analysis' be two key words from an hypothetical mathematical database. B is a more specific term than A in the same field (i.e., B $\subset$A). Suppose that the collection has 2000 documents, 300 of which deal with statistics and only 10 deal specifically with regression analysis. Assume that in these 10 documents the word statistics also occurs. Thus, the vector representing A will have 1's in every position where B does and, therefore, $|A \wedge B| = |B|$. The similarity between A and B, measured by Jaccard's coefficient, is:

$$s_{AB} = \frac{|A \wedge B|}{|A|+|B|-|A \wedge B|} = \frac{|B|}{|A|} = \frac{10}{300} \approx 0.03$$

Hence, the strength of the association between A and B, measured with Jaccard's coefficient, is very small. The situation is essentially the same for the rest of similarity measures mentioned above. The result is that every clustering algorithm using such measures will rarely join A and B in the same cluster, and the association between A and B will therefore not be detected.

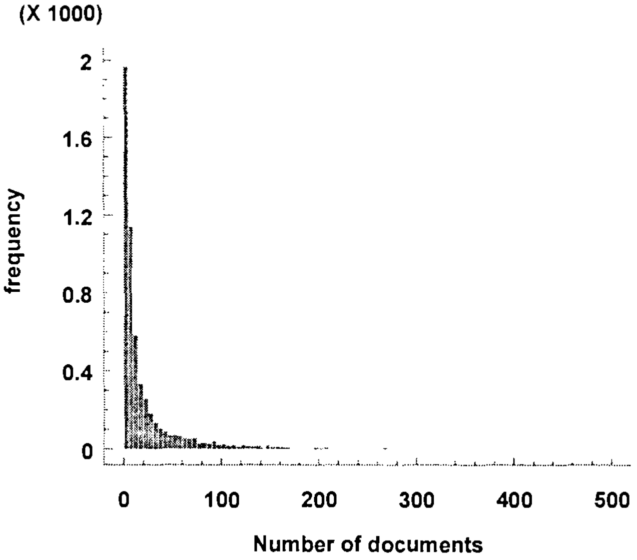

**Fig. 1.** Frequency histogram for terms in the database used in section 4.1. "Number of documents" is the number of documents containing a given word in the document collection. Terms present in more than 500 documents (right side of the histogram) are excluded for the purpose of a better visualization.

If real-valued vectors are used the problem remains, since zeroes in term vectors stay in the same positions. Real-valued similarity measures behave like their binary counterparts, as it is easy to verify.Some additional comments on this subject are made in section 3.3.

The mentioned problems do not arise with documents because of two reasons. Firstly, frequency distributions for documents described by abstracts do not follow Zipf's law, because the number of zeroes in document vectors is by far more homogeneous than for term vectors. Secondly, problems caused by subsethood relations among terms will not occur with documents: there is no sense in saying that a document is a subset of another document. In term space, these relations are possible via the identification of terms to fuzzy subsets introduced in subsection 3.1. This is just the case for descriptors A and B in the preceding example.

The above argument motivates the elaboration of specific algorithms for clustering descriptors which do not suffer the outlined drawbacks. This is done in the next section.

## 3 Compound key words generation

In this section a method for key word generation is proposed. In subsection 3.1 associations between words are modelled using fuzzy set theory. The asymmetric nature of these relations is also noted. In subsection 3.2 a specific architecture to produce semantic classes and extract compound key words is introduced. Finally, in subsection 3.3 alternative methods for producing semantic classes are briefly discussed.

### 3.1 Modelling of term relations using Fuzzy sets

Every descriptor $t_j$ defines a fuzzy subset of documents as follows:

$$D_{tj} = \{ \text{ Relevant documents for term } t_j \}$$

For instance, if $t_j$ = `text databases', $D_{tj}$ is the set made up of documents related to the subject `text databases'. Membership degrees of documents to the $D_{tj}$ sets will vary from 0 (no relation at all with topic $t_j$) to 1 (perfect fitting to the topic). Therefore, it seems appropriate to handle these document sets using fuzzy set theory [27,51]. A few references on fuzzy sets and information retrieval are [5,6,35,37,44].

Given the correspondence between terms and fuzzy sets $t_i \leftrightarrow D_{ti}$ , it is coherent to consider the membership degree of a term to another. Let us numerically define this membership degree by:

$$s_{ij} = \text{degree } (t_i \subset t_j) = |t_i \wedge t_j| \,/\, |t_i| \text{ , where} \tag{7}$$

$$|t_i \wedge t_j| = \sum_{k=1}^{m} min(a_{ki}, a_{kj}) \tag{8}$$

$$|t_i| = \sum_{k=1}^{m} |a_{ki}| \tag{9}$$

When binary term vectors used, $|t_i \wedge t_j|$ is the number of documents in which terms $t_i$ and $t_j$ coincide. In this particular problem, it is not necessary to use the absolute value operator, but it will be used for the sake of coherence with the fuzzy set general theory.

From now on, we will use the function defined in eq. 7 to measure semantic closeness between term vectors. Hence, $s_{ij}$ will denote membership degree of fuzzy subset $D_{ti}$ to fuzzy subset $D_{tj}$. The choice for $s_{ij}$ is a natural one: if two terms $t_i$ and $t_j$ do not coincide in any document, then $s_{ij} = 0$. If $t_i$ occurs in every document where $t_j$ does, then $s_{ij} = 1$. In addition, sets $D_{tj}$ match Kosko's definition of fuzzy subset, that is just now reviewed for the sake of completeness. Given a set C with m elements $c_1, ..., c_m$, every vector $x \in [0,1]^m$ can be viewed as a fuzzy set on C, interpreting vector component $x_i$ as the membership degree of element $c_i$ to the set defined by x. In our case, C is the document set, and term vectors $t_j$ play the role of the x's vectors. To finish off this reasoning, Kosko's fuzzy subset theorem states that membership degrees must be as defined in eq. 7 (for details, see [32].

Function $s_{ij}$ defines two different similarity measures for terms: $s_{ij}$ = degree $(t_i \subset t_j) = |t_i \wedge t_j| / |t_i|$ and $s_{ji}$ = degree $(t_j \subset t_i) = |t_i \wedge t_j| / |t_j|$. In general, $s_{ij} \neq s_{ji}$. The adoption of $s_{ij}$ to model similarities between documents implies the assumption that associations between terms are not symmetric in general. This is a sensible assumption: if we consider for instance the key word `fuzzy set', many people will relate `fuzzy' to `set' stronger than conversely. There are lots of similar examples of this situation. Thus, it is natural to model similarities between terms using two different measures. This asymmetry in term relations has been noted in [10], and is supported by psychological research [2]. It is worth noting that commonly used similarity measures (like Jaccard's coefficient, defined in eq. 5) verify $s_{ij} = s_{ji}$ and, therefore, define a single association measure between terms.

A last remark is in order. If term $t_i$ occurs in every document where $t_j$ does, then $s_{ij}$=1 (maximum membership degree). Thus, it is implicitly assumed that fuzzy subset $D_{ti}$ is entirely contained in fuzzy subset $D_{tj}$, i.e. $t_i \subset t_j$, only as far as our document collection is concerned. If empty words are left out of consideration and the document collection is not very small, this assumption will not cause any trouble, since the probability for two terms to coincide at random in every document where they occur will be very small.

### 3.2 A hierarchical ART architecture for key word generation

In this section we develop a modular term processing system made up of unsupervised ART neural networks. ART stands for Adaptive Resonance Theory [8]. The specific ART model used here is Fuzzy ART [9]. A general scheme of Fuzzy ART architecture is shown in figure 2.

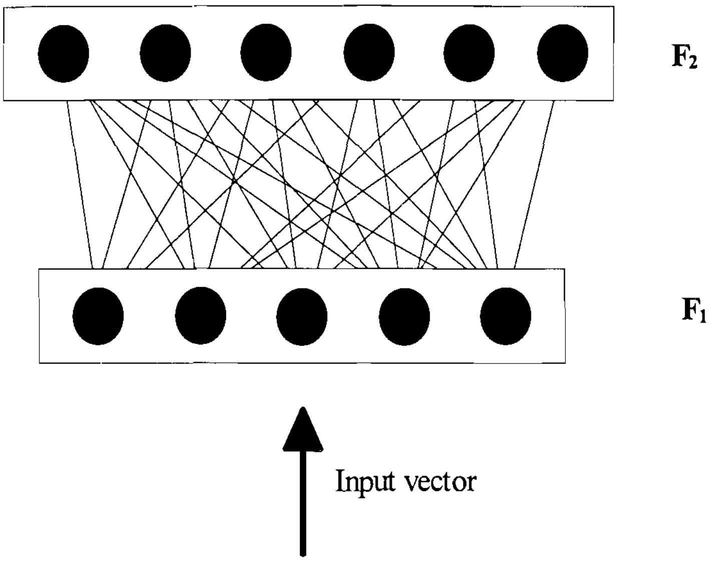

**Fig. 2.** Fuzzy ART architecture

The first layer in figure \ref{fig:idart} is called $F_1$ layer in Fuzzy ART nomenclature. The circles in $F_1$ layer represent nodes of $F_1$. If input vector x lies in $R^n$, then there will be n nodes, node ith containing component $x_i$ of x. Thus, layer $F_1$ acts as a buffer for input vectors. Nodes in $F_2$ layer play the role of clusters in clustering algorithms. The number of $F_2$ nodes is determined in running time, and each node j has an associated vector $V_j \in R^n$, distributed along the connections from that node to nodes in $F_1$ layer. $V_j$ is called prototype vector for cluster jth. A comprehensive exposition of Fuzzy ART model can be found in [9]. There is a relation between unsupervised ART models and clustering algorithms, stated for ART 1 model [8] in [40]. This relation will be used here to explain the operation of Fuzzy ART.

Fuzzy ART makes use of Kosko's point of view about fuzzy sets, via eq. 7. To cluster input vectors Fuzzy ART uses two similarity measures, essentially $s_{ij}$ and $s_{ji}$. The only difference lies in $s_{ij}$, which is replaced by $s_{ij}' = |t_i \wedge t_j| / (\beta + |t_i|)$, where β is a positive constant. There is a control for each measure to determine when a pattern x is too far from any given cluster prototype V. These controls are:

$$\frac{|V \wedge x|}{\beta + |V|} < \frac{|x|}{\beta + n}, \ \beta \in R^+, \ x \in R^n, \text{ for measure } s_{ij}' \qquad (10)$$

$$s_{ji} < \rho, \ 0 \leq \rho < 1, \text{ for measure } s_{ji} \qquad (11)$$

ρ is called the vigilance parameter of Fuzzy ART. The operation of the algorithm is as follows [9,40]:

1. Start with an empty list of prototype vectors (There are no clusters yet).
2. Let x be the next input vector.
3. Find the cluster prototype vector closest to x using $s_{ij}'$. Let V be this vector.
4. If V is too far from x using measure $s_{ij}'$ (or if there are no cluster prototype vectors), then create a new cluster with prototype vector V = x.

5. If V is too far from x using measure $s_{ij}$, deactivate V and go to step 3 to try another prototype.
6. If V is close enough to x according to both measures, then modify V by moving it closer to x. Go to step 2.

Prototype vectors are adapted using the following equation:

$$V^{(new)} = \lambda \, (x \wedge V^{(old)}) + (1 - \lambda) \, V^{(old)} \quad \text{where } 0 < \lambda \leq 1 \qquad (12)$$

Thus, after training, clusters (nodes of $F_2$ using Fuzzy ART terminology) contain patterns that are close enough to each other using the two measures. If Fuzzy ART is used to cluster term vectors, this latter fact means that, given two patterns x and y assigned to the same node, both $x \subset y$ and $y \subset x$ to the extent determined by Fuzzy ART parameters.

There are several aspects of Fuzzy ART that make it an interesting model to cluster terms:

- Fuzzy ART algorithm makes use of two different similarity measures. Thereby, the asymmetric character of term associations is taken into account in a natural manner.
- The number of clusters is automatically determined in running time, without direct human intervention. This is a useful feature because of the specific nature of this problem. First, explorative analysis is complicated by sparsity and high dimension of term vectors. On the other hand, some experiments presented in subsection 3.3 seem to indicate a lack of structure in term spaces. Some comments in the pioneering work of J. Sammon [48] point out in the same direction.
- The particular way in which terms distribute in documents permits to optimize Fuzzy ART algorithm for the task of clustering terms, making it suitable for large database processing. Details are given at the end of this subsection.

Nonetheless, problems mentioned in 2.2, related to term distribution in documents, remain. If a single Fuzzy ART network is used to process the full term set from a database, then two undesirable situations can happen: (1) The most frequent words will form isolated classes for most values of the ART vigilance parameter ρ (including cases when ρ is near zero). (2) If a class gathers both very frequent and rare terms, intersection of all these terms will usually boil down to a non zero component (i.e., a single document in common). Moreover, the grouping of terms in the class will likely be haphazard. We have experimentally verified the occurrence of both situations.

In order to solve these problems, the whole term set T is first divided into frequency groups by descending frequency ordering.

$$T = T_1 \cup T_2 \cup \ldots \cup T_r \qquad (13)$$

Thus, $T_1$ will join the most common terms and, in the opposite extreme, $T_r$ will gather the rare words. Second, new term sets are formed by grouping the Tis as follows:

$T_{12} = T_1 \cup T_2$
$T_{123} = T_1 \cup T_2 \cup T_3$

...

$$T_{123...r} = T_1 \cup T_2 \cup T_3 \cup ... \cup T_r = T \qquad (14)$$

Next, a single Fuzzy ART module is dedicated to each of the $T_i$, i=1,2,...,r-1, and to each of the $T_{12..}$:

$$A_i \rightarrow T_i \text{ ,i=1,...,r-1}$$
$$A_{12} \rightarrow T_{12}$$
$$...$$
$$A_{123...r} \rightarrow T_{123...r} = T \qquad (16)$$

Therefore, the proposed architecture is made up of (r-1) + (r-1) = 2(r -1) single Fuzzy ART modules. In the experimental section r=3 or r=4 will be used, but there is no general rule for partitioning the term set.

Note that no single Fuzzy ART module is dedicated to term set $T_r$, since terms in $T_r$ occur in very few documents and, hence, their vector representations are highly sparse. Use of Fuzzy ART in this situation would lead to the category proliferation phenomenon [9], i.e., the generation of a too high number of clusters (nodes of $F_2$).

Carpenter and Grossberg [9] propose fuzzy complement coding as a solution for this problem, i.e., to replace input vectors x by $X = (x, x^c)$, where $x_i^c = 1 - x_i$. In the present case, fuzzy complement coding is not a solution due to the strong asymmetry between 1's and 0's in term vectors: the fact that two terms have a zero in common (they are both absent from a given document) is not very informative. Therefore, fuzzy complement coding will not be used here. The problem can be minimized by first ranking terms by descending norm. Since $F_2$ prototype vectors initialize to input patterns and these are presented sequentially, prototype vectors will have the largest norms possible from the beginning. Due to the operation of the Fuzzy ART algorithm, unnecessary proliferation of ART classes will be diminished: equation 12 guarantees that a zero vector prototype component will remain zero during training. Hence, prototype norms can only decrease during training. If prototype norms are large at the beginning, they are more likely to be large at the end. In this case, there will be easier for input patterns to fulfill the conditions expressed in eqs. 10 and 11 and, therefore, fewer $F_2$ nodes will be created.

An example of the architecture for r=4 is shown in figure 3.

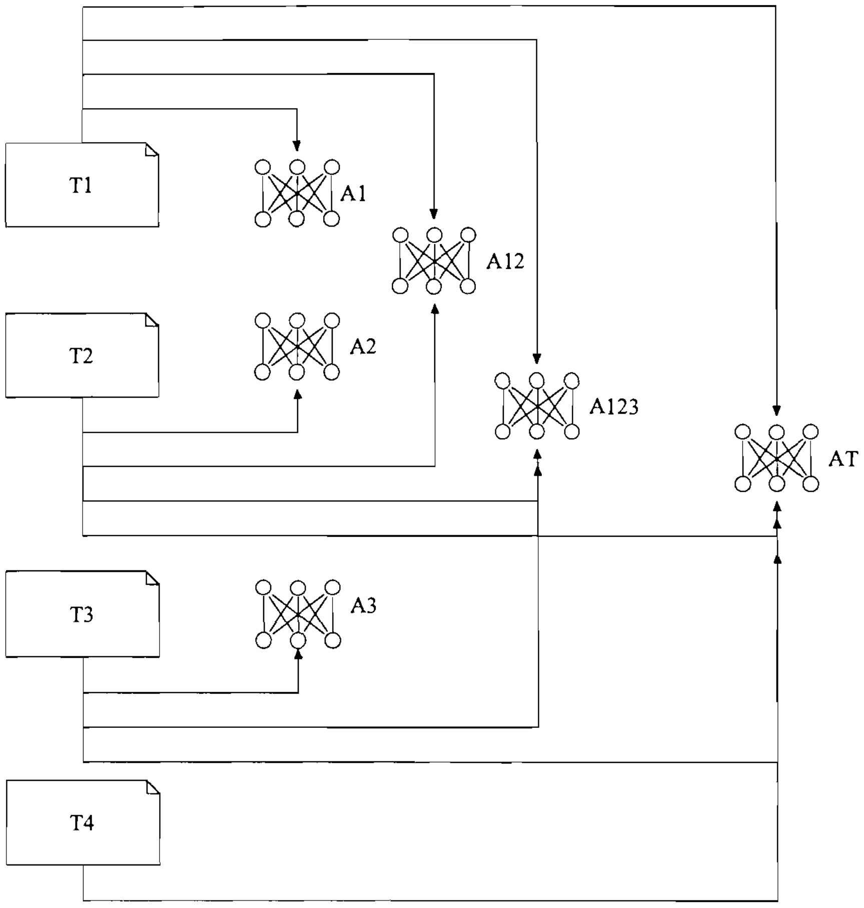

**Fig. 3.** Modular Fuzzy ART based architecture for semantic classes generation

Looking at this figure, we see that a particular term t can belong up to four different semantic classes: one for t ∈ $T_4$, three for t ∈ $T_3$ and four for t ∈ $T_1$ or t ∈ $T_2$. Since key words are formed from words belonging to the same semantic class, common terms will be present in more key words than rare terms.

In order to create term associations to relate each term to every other in the same class is not a good idea; this procedure would produce, in general, too many meaningless associations. We must choose for every term only a few significant related terms. Given a word t ∈ $T_i$ (where $T_i$ can be any of the following: $T_1$ ,$T_2$ , ..., $T_{r-1}$ , $T_{12}$ , ... , $T_{123.r}$ ), we will accept four associations (t, $t_j$) inside class $T_i$ where indexes j are given by:

$$j^*_{1,T_i} = \arg\max_j \frac{|t \wedge t_j|}{|t|} \quad , \quad t_j \in T_i \quad ; \quad j^*_{1,ci(t)} = \arg\max_j \frac{|t \wedge t_j|}{|t|} \quad , \quad t_j \in c_i(t) \quad (17)$$

$$j^*_{2,T_i} = \arg\max_j \frac{|t \wedge t_j|}{|t_j|} \quad , \quad t_j \in T_i \quad ; \quad j^*_{2,ci(t)} = \arg\max_j \frac{|t \wedge t_j|}{|t_j|} \quad , \quad t_j \in c_i(t) \quad (18)$$

where

$c_i(t)$= semantic class of t in Fuzzy ART $A_i$ .

In words, $t_{j*1,Ti}$ is the most closely related broader term for t in T_i; $t_{j*1,ci(t)}$ is the most closely related broader term for t in the semantic class of t (regarding ART $A_i$). Similarly, $t_{j*2,Ti}$ represents the closest narrower term for t in $T_i$, and $t_{j*2,ci(t)}$ the closest narrower term for t within $c_i(t)$. Moreover, $s_{ij}$ gives a measure for the strength of association between terms $t_i$ and $t_j$.

A final computational consideration is called for. As it has been already stated, eq. 12 implies that a zero vector prototype component will remain zero during training. Hence, only non zero prototype components need to be considered during Fuzzy ART operation. Besides this, by Zipf's law, most of the terms in any database are rare terms and, therefore, almost every component of these term vectors will be zero. Since ART prototype vectors initialize their values to input term vectors, vector prototypes sparsity is guaranteed too. Thus, for most of the terms, only a few non-zero components need to be considered for ART calculations. In consequence, increasing the number of documents (and thereby the dimension of the term vector space) has little practical influence in the processing speed. The number of terms is determined by the number of documents in the database, and the ART training time is not considerably affected by this parameter. For instance, the training time for the largest ART, $A_T$, in the database used in subsection 4.1 is only 45 seconds. This collection has 5150 documents and 5800 different terms. Thus, obtaining semantic classes with ART is always a fast process. Processing time rises when similarities between terms have to be computed, because both $s_{ij}$ and $s_{ji}$ have to be calculated for all pairs $(t_i,t_j)$, $i\neq j$. In any case, this is not an insuperable problem: complete process of the above mentioned database does not ever take more than 2 hours in a 125 Mhz workstation (while other users are running their own programs). This time is quite acceptable for this type of tasks, because no document database require continuous updating. In summary, the proposed ART architecture is quite suitable for processing relatively large databases. The only handicap when handling a very large database could be the disk space required to store similarities $s_{ij}$ and $s_{ji}$ during the process of key words generation. In any case, with actual disk storage capacities, this seems not to be a problem.

### 3.3 Other methods for semantic class formation

An interesting question arises when we ask about the existence of structure within term vector space. That is, given a document database, do base terms organize into clusters according to their semantic meanings? Should this be the case, we could employ various extensively used clustering algorithms to form term groups. These groups could then be identified with semantic classes, in the sense used throughout this paper. To experimentally test this hypothesis, three types of unsupervised clustering algorithms were considered:

- Hierarchical clustering algorithms.
- Non hierarchical algorithms, such as k-means and Self Organizing Maps (SOMs).
- Hybrid not supervised and non-hierarchical algorithms such as Fuzzy c-means algorithm.

A handicap in using hierarchical clustering algorithms is that similarity measures used by these algorithms are symmetric: $s_{ij} = s_{ji}$. Single link and complete link methods were used on different test databases, including databases of section 4. Clusters produced by

these methods are very small (usually two elements), and very common terms tend to form their own isolated clusters. If associations have to be formed from words in the same semantic class, the result will be that general terms will not attain any association! The situation is similar to that produced when using a single Fuzzy ART module on the whole term set, with a high vigilance parameter ρ.
This result seems to indicate the lack of structure in document space, and it is coherent with the work of R. Burgin [7]. In his paper Burgin notes that hierarchical algorithms tend to produce many clusters of documents with only two elements.

On the other hand, we have extensively used Kohonen Self Organizing Maps (SOMs) on term and document collections to visualize the structure of the very high dimensional spaces in which they lie. The basic adapting structure in a SOM is a (usually 2D) network of interconnected neurons, each endowed with an associated n-dimensional *pointer*. We denote the connectivity pattern and the set of pointers as τ and W respectively. In the 2D case, τ is usually based on the square or the hexagon, in which case neurons not lying on edges of the network have direct links with 4 and 6 other neurons respectively.Pointers are subject to learning as input vectors are presented to the network. The set of pointers after the t-th presentation is denoted by
W(t)= { w(i,j)(t)}; The final set of pointers is denoted as W(T) = {w(i,j)}. Further details can be found in [29,30].

In [41,43] a technique is developed to visualize pointers of SOMs in two dimensions. Specifically, we consider the *median-interneuron-distance* or *MID* matrix as that whose (i,j) entry is the median of the (Euclidean) distances between w(i,j) and all pointers in a neighborhood $N_D$(i,j) [31,33,43]. The MID entries can be converted to gray levels for a better visualization. In these images, light zones indicate large distances between neuron pointers. Thus, clusters in data can be identified as dark zones surrounded by lighter units.

In the following experiment, two document databases were considered. The first collection has 1869 documents covering six main topics: `archives preservation', `mathematics, statistics and regression', `language comprehension, reading, teaching', `special libraries', `chemical nomenclature' and `database management systems'. The database is extracted from ERIC database (see subsection \ref{subsec:diez} for an ERIC system description), and there are 3427 different terms. The second database has 502 documents from ERIC database, covering eight topics: `automatic indexing', `transformational generative grammar', `Prolog', `paper preservation', `bayesian statistics and mathematics', `artificial intelligence and psychology', `radiology' and `sport psychology'. There are 1460 different terms.
In figure 4 MID matrices for the two term sets are displayed. Figure 4(a) is very similar to that obtained when displaying uniform random data (a figure illustrating this latter case can be found in [33]). Figure 4(b) reveals more structure, because the eight topics chosen are completely disjoint. In any case, it is very hard to say anything conclusive by only looking at the figure. Manual checks of term distribution in the SOMs images do not reveal the existence of clearly defined clusters.

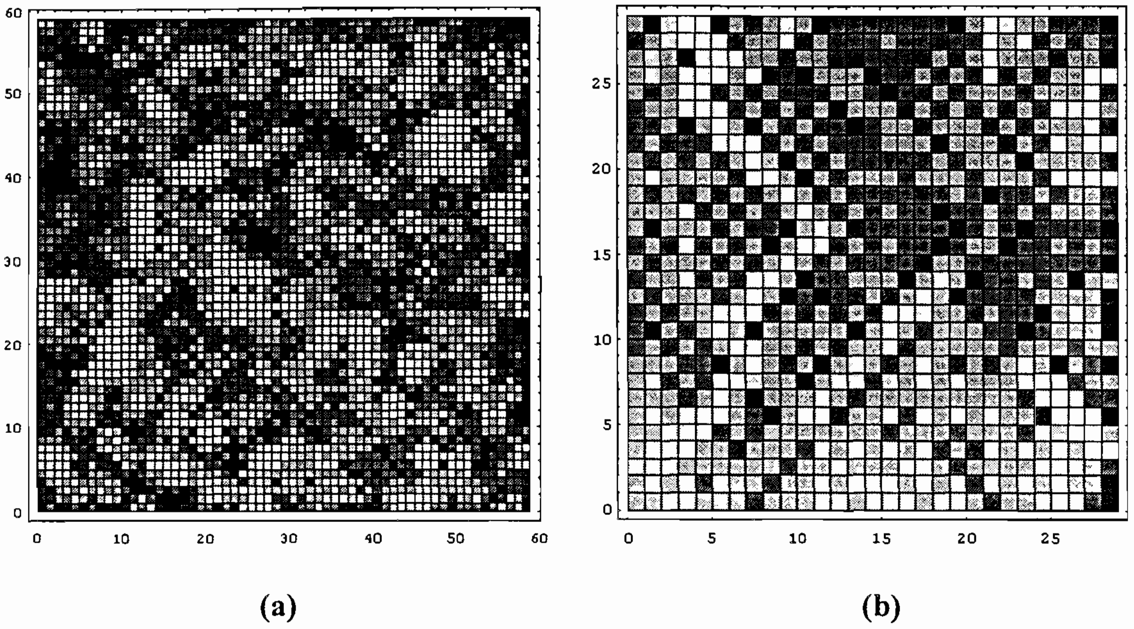

**Fig. 4.** SOM gray-level image for two different IR databases. (a) Database with 6 different topics. (b) Database with 8 different topics.

These gray-level images are very different when data have a clearly defined cluster structure. Figure 5 displays the scatter plot and the MID matrix for 200 points in R$^2$, well distributed in three clusters. The structure in figure 5 is apparent. Dark rectangles indicate short distances between neuron pointers. Lighter rectangles indicates larger distances. Thus, figure 5(b) shows two close clusters, and a third, more distant.

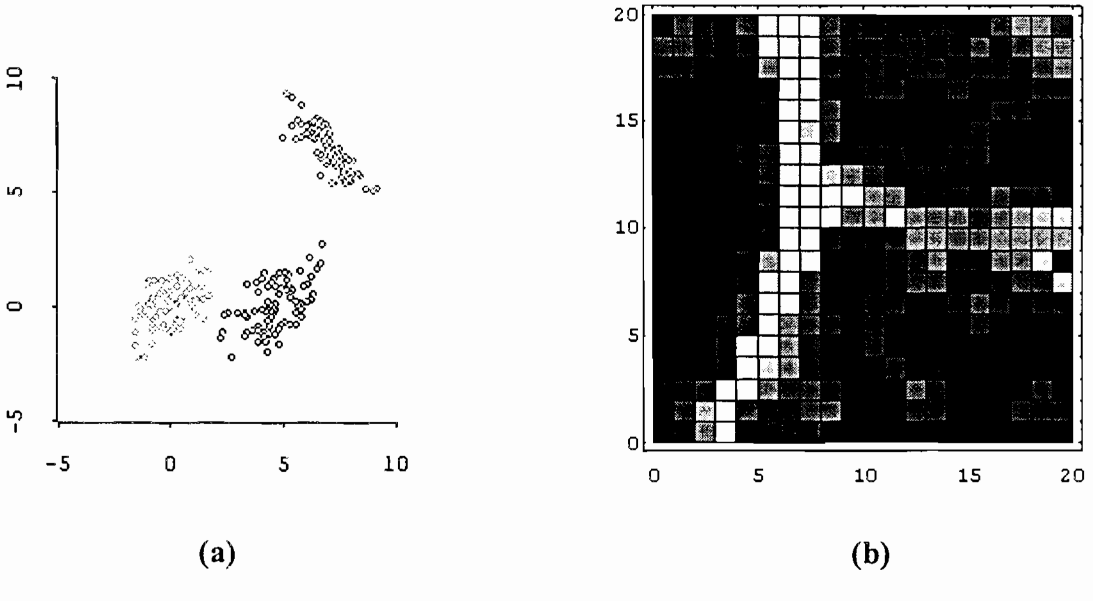

**Fig. 5.** SOM gray-level image for 3 clusters in R$^2$. (a) Scatter plot. (b) SOM gray-level based image.

Figure 6 shows a data set with 500 points distributed into five clusters with different covariance structures, and some outliers. Figure 6(a) shows the Sammon's mapping \cite{bib:sammon69} to R$^2$ for this data set. There are six outliers, five of them identified with crosses and the sixth with a solid dot. Figure 6(b) clearly shows the cluster structure in the data set.

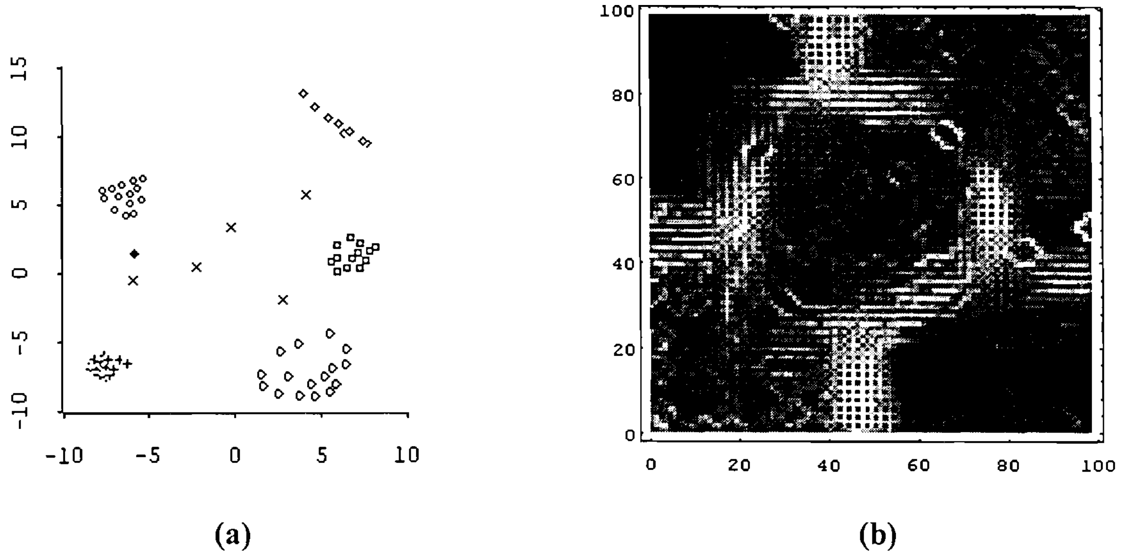

**Fig. 6.** SOM gray-level image for 5 clusters in $R^{10}$. (a) Sammon mapping in $R^2$. (b) SOM gray-level based image.

Therefore, we are inclined to believe that there is no cluster structure in document space, neither in term space.

Other overlapping clustering algorithms, such as Fuzzy c-means algorithm [3], use membership degrees to cluster input vectors. These membership degrees are known to approximate the a posteriori membership probabilities of terms to classes [4], when clusters are spherical and well separated. By a spherical cluster we mean that the data covariance matrix has the form $\Sigma^2 = \sigma^2$ I. Therefore, membership degrees will not be very informative when this is not the case. We think it is not advisable to use fuzzy clustering algorithms for terms represented like here.

## 4 Experimental work

Before getting involved in any concrete experiment it is convenient to make some observations about the way to evaluate outcomes of such experiments.

A thesaurus can be evaluated by using a query set and measuring retrieval effectiveness for that query set when the thesaurus is used and when it is not used. This has been done over the years by means of *recall* and *precision* measures. Assuming a query has been formulated, recall can be defined as the proportion of relevant documents retrieved and precision as the proportion of retrieved documents that are relevant [46]. The origin of these measures lies in the pioneering IR research in the 60s: ASTIA [23], Cranfield I [12,13,34] and Cranfield II [14,15]. Already in these early experiments a trade-off was detected between recall and precision: if recall is too improved, precision worsens and conversely. This inverse relation has been corroborated in recent studies [18,21,24]. An additional problem when using relevance and precision measures is the need to know which documents are relevant for queries formulated to the database. However, *relevance* is a subjective concept, as it is shown in the experiments by Lesk and Salton

[36] and Gomez [20]. These works demonstrate that it is not unusual the situation in which two experts in the same field do not agree in which documents are relevant for a given query. On the other hand, it is not possible to have queries available (and knowledge of relevant documents for them) for all potential situations. Furthermore, results obtained could depend on the particular retrieval engine used. These objections stand for manual thesauri as well.

Therefore, we are not going to make use of a test query set and recall and precision measures to evaluate automatically obtained word associations. Rather, the approach used here is to work with databases for which good manual thesauri are available and thus, to study the quality of automatic term associations with respect to such manual thesauri.

### 4.1 Information Retrieval database from ISA

This database contains 5150 documents on the subject `information retrieval', retrieved from the commercial CD-ROM database ``Information Science Abstracts Plus" (from Silver Platter Information Inc). Articles date from 1966 to June 1993. For an example of the structure of the records used in this experiment, see table 1. TI stands for title, DEM stands for mayor descriptors, DER for minor descriptors, and AB for abstract. DER and DEM descriptors are used to join documents on similar topics. Major descriptors denote primary topics. Both DER and DEM descriptors are chosen from ERIC thesaurus (by the creators of the CD-ROM database). ERIC thesaurus is associated to ERIC database and it is briefly described in the next section, where a document set from ERIC database is used.

**TI:** Multiversion Information Retrieval Systems and Feedback with Mechanism of Selection.

**DEM:** *Algorithms-; *Information-Retrieval

**DER:** Comparative-Analysis; Feedback-; Mathematical-Formulas; Relevance-Information-Retrieval; Selection-; User-Needs-Information; User-Satisfaction-Information

**AB:** Discusses the design of multiversion information retrieval systems and provides a theoretical justification for the necessity of creating such systems to perform an optimal search for the user's information needs. Topics discussed include comparing query formulations; feedback algorithms; an experiment with a test collection; and the mechanism of selection. (20 references).

Table 1. Example of ISA database record

For each document, only its title and abstract are considered for automatic processing. ERIC descriptors are removed so the results cannot be altered in any way by manual descriptors.We want to test the ability of the proposed ART-based system for detecting the foremost relationships in the IR field. Examples of these relevant associations are `information retrieval', `information science', `search strategies', and others.

The number of different single words in the collection is 13670. After using an empty word list (the same for all experiments) and removing words that occur only in one or two documents, there are left 5800 single descriptors. Using the space vector model to represent these descriptors, we have a collection of 5800 term vectors in $R^{5150}$.

As explained in subsection 3.2, the whole term set has to be divided into r frequency regions. As pointed out in section 2, there is no general rule to do this. In preliminary work on a smaller database [42] we used r=3 so that, using the notation of subsection 3.2, $T_1$ joined very frequent terms, $T_2$ joined moderately frequent terms, and $T_3$ joined rare words. This time we shall use r=4, splitting the very frequent term group $T_1$ in two, for the sake of a better frequency resolution.

The frequency cutpoints are chosen so that, roughly, most frequent terms ($T_1$) correspond to the 5-upper percentile term frequency distribution, very frequent terms ($T_2$) correspond to the 15-upper percentile, moderately frequent terms ($T_3$) to the 50-upper percentil, and the rest (50%) are considered rare terms. Table 2 shows the frequency distribution for this database. For instance, the first column of table 2 shows that there are 291 terms in the very frequent term class $T_1$, they represent a 5% of the whole term set T, and each of these terms occurs in at least 139 documents.

We have carried out some other similar experiments using different frequency partitions of T using r=3 and r=4. If the test database is relatively small (it has no more than 1000 documents, say), there are not significant differences. But if the database is not too small (larger than 1000 documents, say), better results are obtained for r=4. The explanation is simple: if the database is small, then the number of index terms will be small, and there is no significant difference in splitting the very frequent term class. But if the index term set is large, there are appreciable differences between the most frequent term set $T_1$ and the very frequent term set $T_2$.
In this latter case, there is not too much difference in the following frequency partitions: 5%-10%-35%-50% (used here), 5%-10%-30%-55% and 5%-10%-25%-60%; that is, results are similar for all these partitions.

| Term sets | $T_1$ | $T_2$ | $T_3$ | $T_4$ |
|---|---|---|---|---|
| % | 5% | 10% | 35% | 50% |
| Total | 291 | 588 | 1962 | 2959 |
| Frequency of term | ≥ 139 | [53,159) | [10,53) | [3,10) |

**Table 2.** Frequency partition for ISA database

In all experiments made in this work, ART parameters are set to $\lambda$=1 (fast learning in ART terminology), $\beta$=0 in control devices for similarity measures, and $\rho$ = 0.001, that is, a very low vigilance paramenter, so that the system can stabilize at the minimal attainable number of classes. In fast learning mode, Fuzzy ART needs only two cycles over the whole term set. Neither the number of Fuzzy ART nodes nor the prototype vectors will change in successive cycles. After every Fuzzy ART module has been trained, associations are made as explained in section 3.2, and joined together in a common database.

Table 3 shows the relationship between the number of terms in a frequency class and the number of nodes of the corresponding ART module.

| Term set | Number of terms | ART module | Number of nodes |
|---|---|---|---|
| $T_1$ | 291 | $A_1$ | 31 |
| $T_2$ | 588 | $A_2$ | 84 |
| $T_{12} = T_1 \cup T_2$ | 879 | $A_{12}$ | 84 |
| $T_3$ | 1962 | $A_3$ | 356 |
| $T_{123} = T_1 \cup T_2 \cup T_3$ | 2841 | $A_{123}$ | 349 |
| $T_{1234} = T_1 \cup T_2 \cup T_3 \cup T_4$ | 5800 | $A_{1234}$ | 979 |

**Table 3.** Term subsets by frequency and the corresponding ART modules

As it is readily seen in this table, the rise in the number of ART nodes is not only due to a higher number of terms (compare $A_2$ to $A_{12}$ or $A_3$ to $A_{123}$), but also to the increasing sparsity of term vectors. Furthermore, the relation between the number of terms in a frequency term class and the number of nodes of the corresponding ART module is not linear. Rather, the hierarchical division of the term set is reflected by the number of classes in the corresponding Fuzzy ART modules. This seems to indicate the adequacy of the proposed ART hierarchical model to cluster the particular vectors used here. Terms in $T_3$ are more sparse than terms in $T_1$ or $T_2$. This is the cause of the strong rise on the number of ART nodes, as table 3 shows. This fact corroborates our caution of not employing a dedicated ART moduled on the rare term class ($T_4$) alone.

An odd fact is that increase in the number of terms do not always gives rise to an increment in the number of ART nodes (compare $T_2$ to $T_{12}$ and $T_3$ to $T_{123}$). As pointed out at the end of section 3.2, ART classes prototypes initializes with the highest norm input vectors. Therefore, initial prototypes for nodes of $A_{12}$, for instance, will have less zeroes than prototypes in $A_2$, and this is an stabilizing mechanism that prevents the category proliferation phenomenon, as expected.

Next, the percentage of key words from ERIC thesaurus present in the set of automatically extracted key words is computed. Single key words, as `simulation' for instance, will not be taken into account in the sequel because their production is not related to the method under study, but previous to it. The proportion of single key words from ERIC present in the automatic term set is about 95% for the databases considered in this study. This is not surprising since the bulk of ERIC single terms are important words (seldom they are rare terms) and the automatic method does not reject them. Of course, some of the compound key words are not detected due to the lack of any of the single terms that form them in the automatic term set.

There are no fixed rules for assigning key words to documents in general; different authors may ascribe different descriptors for the same document. This fact implies that not all descriptors must be considered to have the same relevance. Thus, we will only consider descriptors used by several authors. The minimum threshold is set here to three, since we have excluded from the database vocabulary words occurring in only one or two documents. Coincidence of experts in the use of the same descriptor guarantees its validity and generality of use.

Within ERIC descriptor set three subgroups will be considered: high frequency terms, used by at least ten authors; medium frequency terms, used by at least five authors and by no more than nine; and at last low frequency terms, used by three or four authors. The most relevant set is the high frequency term set, but it is interesting to see what happens to the two other categories. The results for DER descriptors are in table 4 and for DEM descriptors in table 5. One thing that can happen is that in a three-word term only two associations (of three) are detected. For instance, for the term `junior-high-schools' our system detected the associations `junior-high' and `high-school'. We label this situation as `relaxed detection' in the tables. `Strict detection' happens when every possible pair of associations is detected, and this is the rule followed here to produce three and four word relations.

| Match | High frequency terms | % | Moderate frequency terms | % | Low frequency terms | % |
|---|---|---|---|---|---|---|
| Strict | 86 | 60% | 48 | 48% | 35 | 26% |
| Relaxed | 97 | 68% | 57 | 57% | 44 | 33% |
| Total | 140 | 100% | 100 | 100% | 133 | 100% |

**Table 4.** Detected very frecuent compounds key words (DER descriptors)

| Match | High frequency terms | % | Moderate frequency terms | % | Low frequency terms | % |
|---|---|---|---|---|---|---|
| Strict | 128 | 62% | 57 | 33% | 29 | 15% |
| Relaxed | 138 | 67% | 69 | 40% | 46 | 24% |
| Total | 205 | 100% | 172 | 100% | 188 | 100% |

**Table 5.** Detected very frecuent compounds key words (DEM descriptors)

We see that for both DER and DEM descriptors, the proportion of manual assigned descriptors recovered is never below 60%. This is a remarkably satisfactory rate in any case, and it is worth noting that the proposed method uses only document coocurrence information (not contiguity information in sentences, for example). The drop in the hit rate for low frequency terms is not surprising, since the automatic method is based on frequency statistics and looks only for the strongest (more frequent) associations -- words occurring only in three documents cannot have strong associations.

There are some other considerations that add value to the proposed key word generation system:

(a) The method is able to detect many relations between non adjacent words. That is the case of descriptor `recall -precision' (not present in the manual thesaurus).

(b) For terms which manually-assigned relations are not detected, other similar ones are. A few examples: `elementary education', `decision making' or `state agencies' are not detected, but `elementary school', `decision theory' and `government agencies' are given instead.

(c) The method detects significant relations for relevant terms that the manual thesaurus fails to include. A few examples are: OPAC (`OPAC guide', `OPAC access'), keyword (`keyword searching', `keyword efficiency'), MARC (`MARC records', `MARC format'), query (`query relevance', `automatic query'), term (`term index', `term descriptor'), fuzzy (`fuzzy sets', `fuzzy logic'), boolean (`boolean logic', `boolean search'), probabilistic (`probabilistic ranking', `probabilistic estimation').

(d) For some terms, the manual thesaurus offers only non-specific relations: `logic philosophy', `logic thinking'. The automatic method detects context-dependent relationships: `boolean logic', `logic skills'.

(e) We have a numerical measure for the strength of each association, $s_{ij}$. This membership degree of topic represented by term $t_i$ to topic represented by term $t_j$ has a nice property: Often, descriptors offered by the manual thesaurus have the highest association degree. One example: (cable → satellite, 0.28), (satellite →cable, 0.31), (cable → television, 0.76). The last one, `cable television' is the only present in manual thesaurus. A low membership degree does not always mean that a relation must be rejected, though. For example, `administrative-policy', a rare manual descriptor in this database, is present in the automatic database with degree 0.09. Thus, in this paper we do not remove relations under a preespecified membership threshold value.

To conclude this section some particular cases of associations are shown, so that the reader can complete his perception of the way the method works.

Next we show some strong relationships for the term `retrieval':

(information → retrieval, 0.57), (retrieval → information, 0.69), (text → retrieval, 0.62), (document → retrieval, 0.67), (indexing → retrieval, 0.50), (evaluation → retrieval, 0.57), (language → retrieval, 0.55), (query → retrieval, 0.68), (system → retrieval, 0.37), (storage → retrieval, 0.76), (probabilistic → retrieval, 0.79).

The symbol → indicates that the first term is a subset, or narrower term, of the second. It is worth noting that `retrieval' corresponds to a very general topic: its only broader term is `information'. We can see that `information' is a subset of `retrieval' too, but to a lower degree. In this way, an implicit hierarchy is defined between descriptors.

Next we show some associations for term `indexing', a more specific term than `retrieval' in this database:

(automatic → indexing, 0.36), (documents → indexing, 0.12), (indexing → documents, 0.13), (problems → indexing, 0.10), (process → indexing, 0.11), (project → indexing, 0.12), (indexing → retrieval, 0.50), (controlled → indexing, 0.39), (record → indexing, 0.17), (selective → indexing, 0.18), (coordinate → indexing, 0.80), (dictionary → indexing, 0.25), (abstracting → indexing, 0.60), (consistency → indexing, 0.43), (indexers → indexing, 0.57), (indexer → indexing, 0.50).

Next we show associations for term `MARC', a very specific term:

(MARC → congress, 0.31), (MARC → records, 0.56), (MARC → cataloging, 0.22), (MARC → library, 0.62), (Dewey → MARC, 0.31), (union → MARC, 0.20), (council → MARC, 0.20).

The fact that associations where `MARC' occurs on the right side (i.e., MARC is the main term) have a low degree (0.20), indicates that `MARC' is a very specific term.

Next, an example of the system ability for detecting relations with syntactic variants:

(search → online, 0.35), (online → search, 0.41), (online → searching, 0.38), (searchers → online, 0.66), (presearch → online, 0.01).

Just another example: for the manual thesaurus key word `library catalog', the automatic system associates library to the following variants: catalog, cataloging, catalogs, catalogue, catalogues, cataloguing (with slightly varying degrees).

The manual thesaurus also admits some syntactic variants, as `library schools' and `schools libraries', but there are only a few key words in this situation.

Many of the associations are reproduced in various ART modules. For instance, `information retrieval' is detected in every ART module where `information' and `retrieval' are considered ($A_1$, $A_{12}$, $A_{123}$ and $A_T$). But some of the word associations are detected only in one of the ART modules of the system. This fact indicates that all the modules are useful such as they are used.

In general, a single word has a maximum possible number of relations, depending on the number of ART modules used. But this fact does not mean that every word will attain this maximum number of relations, due to the repeated associations. In this way, the system automatically regulates the number of associations, and this number does not need to be imposed by hand. On the other hand, it would not be fair to allow every word in the database to have the same number of relations. In this way, words describing major topics, such as `information', have a higher number of associations, and tend to occur at the right place of the associations.

To summarize this example, the proposed automatic system is capable of detecting a high percentage of the manual thesaurus key words (elaborated from a huger corpus), and detects many other significant relations between single terms. Besides this, syntactic variants of the same key word are automatically detected. In addition, a numerical degree of the strength of the association between single term in compound key words is given. This degree could help users to move through document database in networked information environments.

## 4.2 Another multitopic database from ERIC

In this experiment we are going to use a database extracted from ERIC system. The ERIC database is an information system sponsored by the U.S. Department of Education, formed by two main sources: the ``Resources in Education" (RIE) file of document citations and the ``Current Index to Journals in Education" (CIJE) file of journal article citations from over 750 professional journals. Our database is formed from ERIC database using the query:

(acoustics in DE) OR (alcoholism in DE) OR (astronomy in DE) OR (bayesian in DE) OR (cognitive-psychology in DE) OR (computer-games in DE) OR (genetics in DE) OR (military-service in DE) OR (neurolinguistics in DE) OR (water-pollution in DE)

where a document will be retrieved if any of the preceding key words are present in its DER or DEM descriptors. The number of retrieved records is 1054. There are 8820 different words, and after using an empty word list (the same that in the preceding database) and removing words occurring only once or twice, as in the preceding database, there are left 4501 single terms.

| Term sets | $T_1$ | $T_2$ | $T_3$ | $T_4$ |
|---|---|---|---|---|
| % | 5% | 10% | 31% | 54% |
| Total | 224 | 441 | 1400 | 2436 |
| Frequency of term | $\geq 36$ | [14, 36) | [4, 14) | [2, 4) |

**Table 6.** Frequency partition for ERIC database

| Term set | Number of terms | ART module | Number of nodes |
|---|---|---|---|
| $T_1$ | 224 | $A_1$ | 21 |
| $T_2$ | 441 | $A_2$ | 68 |
| $T_{12} = T_1 \cup T_2$ | 665 | $A_{12}$ | 61 |
| $T_3$ | 1400 | $A_3$ | 211 |
| $T_{123} = T_1 \cup T_2 \cup T_3$ | 2065 | $A_{123}$ | 214 |
| $T_{1234} = T_1 \cup T_2 \cup T_3 \cup T_4$ | 4501 | $A_{1234}$ | 515 |

**Table 7.** Term subsets by frequency and the corresponding ART modules. ERIC database.

Our aim here is to simulate the situation in which information is received from many different sources in small amounts, and the task is again to study the proportion of manual descriptors retrieved.Unlike the preceding database, this small collection contains few documents on each topic, so that it will be harder to detect significant word associations by only using frequency information. Besides this, manual key words are produced using a far larger amount of information than it is available in this toy database. Despite these objections, we think this is an interesting experiment to carry out.

Terms are joined into frequency groups as in the preceding ISA database. Term frequency groups are shown in table 6. Statistics for the ART system are shown in table 7. Similar remarks than for ISA database apply for this database.

Manual descriptors for this database are joined in three frequency classes as in the preceding example. Detection statistics for DER and DEM descriptors are shown in tables 8 and 9.

| Match | High frequency terms | % | Moderate frequency terms | % | Low frequency terms | % |
|---|---|---|---|---|---|---|
| Strict | 37 | 51.4% | 38 | 36.2% | 37 | 26.4% |
| Relaxed | 42 | 58.3% | 46 | 43.8% | 51 | 36.4% |
| Total | 72 | 100% | 100 | 100% | 140 | 100% |

**Table 8.** Detected compound very frecuent compounds key words for ten-topic database (DEM descriptors)

| Match | High frequency terms | % | Moderate frequency terms | % | Low frequency terms | % |
|---|---|---|---|---|---|---|
| Strict | 44 | 44% | 53 | 35.1% | 42 | 21.4% |
| Relaxed | 53 | 53% | 59 | 39% | 52 | 26.5% |
| Total | 100 | 100% | 151 | 100% | 196 | 100% |

**Table 9.** Detected compound very frecuent key words for ten-topic database (DER descriptors)

Detection rates for this 1000 document database are worse than those for the ISA database, as expected. In any case, for the most important descriptor set, detection rates are over 50%, and this is a good result, given the minimum information used to obtain it.

To end this example, a singular long key word automatically detected: `acquired-immune- deficiency- syndrome'. This key word is also present in ERIC thesaurus.

## 5 Conclusions

In this paper the task of compound key word generation from a document collection has been addressed. For each document, only title and abstract are taken into account. This information is available for documents in most commercial information systems, and for documents in Internet as well. Basic inputs are the occurrence of words in documents.

The work is carried out in two steps: first, a hierarchy of ART networks is used to automatically generate key word lists (called semantic classes here) from the document database. Second, important relations between words in the same semantic class are detected, using fuzzy subsethood measures. In this way, the risk of considering meaningless associations is highly reduced. The search space for relations is strongly reduced too. Frequent words achieve more semantic relations than rare words.

Relations obtained are asymmetric, as in natural language happens. Every relation has associated a numerical measure of its strength, that can be very useful for users to move through the vocabulary of the database. Given a word, its strongest associations are that with highest association measures. But the measure is not valid to compare significance of key words without common words: rare words have always low measures.

About the way of obtaining semantic classes, the experiments carried out seem to show the lack of structure in document space. This fact advise us against the use of more established clustering algorithms, that presuppose this structure to exist.

To evaluate the automatic key word set, document collections for which a manual thesaurus exists are used. Results are better for large document collections than for small ones. This result is foreseeable, since only frequency of words in documents is used. The recall rate for the important manual descriptors never goes under 50%, even in small databases. It is worth noting that, unlike the manual thesaurus, the automatic system has only available the information of a few thousand documents to generate the key words.

A last remark is in order: the techniques presented here should not be straightforwardly extended to process full text databases. If it was done so, some noise would be introduced in the system: not all words in a long document have to be related with each other. A possible solution could be to partition the text of the document in smaller sections.

**Acknowledgements.** The author wishes to thank the support from the DGICYT PB94-0374 Spanish grant. The author is also grateful to two anonymous referees for their valuable comments and remarks.

## References

1. M.R. Anderberg. *Cluster Analysis for Applications*. Academic Press, New York, 1976.
2. J.R. Anderson. *Cognitive psychology and its Implications*. Freeman, New York, second edition, 1985.
3. J.C. Bezdek. Pattern recognition with fuzzy objective function algorithms. Plenum, New York, 1981.
4. J.C. Bezdek, R.J. Hathaway, and V.J. Huggins. Parametric estimation for normal mixtures. PatternRecognition Letters, (5):79-84, March 1985.
5. D.A. Buell. An analysis of some fuzzy subset applications to information retrieval systems. Fuzzy Sets and Systems, 7:35-42, 1982.
6. D.A. Buell. A problem in information retrieval with fuzzy sets. Journal of the American Society for Information Science, 36(6):398-401, 1985.
7. R. Burgin. The retrieval effectiveness of five clustering algorithms as a function of indexing exhaustivity. Journal of the American Society for Information Science, 46(8):562-572, 1995.
8. G.A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics, and Image Processing, 37:54-115, 1987.
9. G.A. Carpenter, S. Grossberg, and D.B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks, 4:759-771, 1991.
10. H. Chen and K.J. Lynch. Automatic construction of networks of concepts characterizing document databases. IEEE Transactions on System, Man and Cybernetics, 22(5):885-902, Sept.-Oct. 1992.

11. D.B. Cleveland and A.D. Cleveland. Introduction to Indexing and Abstracting. Libraries Unlimited, Inc., Littleton, Colorado, 1983.

12. C.W. Cleverdon. Report on the first stage of an investigation into the comparative efficiency of indexing systems. Technical Report, College of Aeronautics, Cranfield, 1960.

13. C.W. Cleverdon. Report on the testing and analysis of an investigation into the comparative efficiency of indexing systems. Technical Report, College of Aeronautics, Cranfield, 1962.

14. C.W. Cleverdon. The Cranfield tests of index language devices. Aslib Proceedings, 19:173-194, 1967.

15. C.W. Cleverdon. Design and evaluation of information systems. Annual Review of Information Science and Technology, 6:42-73, 1971.

16. C.J. Crouch. An approach to the automatic construction of global thesauri. Information Processing and Management, 26(5):629-640, 1990.

17. J. December. New spiders roam the web. Computer-Mediated Communication Mgazine, 1(5), 1994.

18. D. Ellis. New horizons in information retrieval. The Library Association, London, 1990.

19. B. Everitt. Cluster analysis. Gower, Halsted Press, New York, 2nd Edition, 1981.

20. L.M. Gomez, C.C. Lochbaum, and T.K. Landauer. All the right words: finding what you want as a function of richness of indexing vocabulary. Journal of the American Society for Information Science, 37(1):3-11, 1986.

21. M.D. Gordon and M. Kochen. Recall-precision trade-off: a derivation. Journal of the American Society for Information Science, 40(3):145-151, 1989.

22. A. Griffiths, H.C. Luckhurst, and P. Willet. Using interdocument similarity information in document retrieval systems. Journal of the American Society for Information Science, 37(1):3-11, 1986.

23. C.D. Gull. Seven years of work on the organization of materials in the special library. American Documentation, 7:320-329, 1956.

24. S.P. Harter. Search term combinations and retrieval overlap: A proposed methodology and case study. Journal of the American Society for Information Science, 41(2):132-146, 1990.

25. A.K. Jain and R.C. Dubes. Algorithms for clustering data. Prentice Hall, Englewood Cliffs, NJ, 1988.

26. Y. Jing and W.B. Croft. An association thesaurus for information retrieval. Technical Reprot TR-93-026, Department of Computer Science, University of Massachusetts at Amherst, May 1993.

27. G.J. Klir and T.A. Folger. Fuzzy sets, uncertainty and information. Prentice-Hall International, Inc., New Jersey, 1988.

28. T.R. Kochtanek. Document clustering, using macro retrieval techniques. Journal of the American Society for Information Science, 34(5):356-359, 1983.

29. T. Kohonen. The Self-Organizing Map. Proceedings of the IEEE, 78(9):1464-1480, 1990.

30. T. Kohonen. Self-Organizing Maps. Springer Verlag, Heidelberg, 1995.

31. T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. SOM_PAK. The self-organizing map program package. Technical Report, Helsinki University of Technology, Laboratory of Computer and Information Science. Espoo, Finland, March 1995.

32. B. Kosko. Neural networks and fuzzy systems: A dynamical approach to machine intelligence. Prentice Hall, Englewood Cliffs, New Jersey, 1991.

33. M.A. Kraaijveld, J. Mao, and A.K. Jain. A nonlinear projection method based on Kohonen's topology preserving maps. IEEE Transactions on Neural Networks, 6(3):548-559, 1995.

34. F.W. Lancaster and J. Mills. Testing indexes and index language devices. American documentation, 15:4-13, 1964.

35. H.L. Larsen and R.R. Yager. The use of fuzzy relational thesauri for classificatory problem solving in information retrieval and expert systems. IEEE Transactions on Systems, Man and Cybernetics, 23(1):31-41, 1993.

36. M.E. Lesk and G.Salton. Relevance assessments and retrieval system evaluation. Information storage and retrieval, 4:291-303, 1968.

37. D. Lucarella and R. Morara. FIRST: Fuzzy Information Retrieval System. Journal of Information Science, 17(1):81-91, 1991.

38. B. Mandelbrot. An information theory of the statistical structure of language. In Proceedings of the Symposium on applications of communication theory, pages 486-500, Butterworth, London, 1953.

39. B. Mandelbrot. On the language of taxonomy: an outline of a thermostatistical theory of systems of categories with willis (natural) structure. In Information theory: Papers read at a Symposium on information theory, pages 135-145, Butterworth, London, 1956.

40. B. Moore. ART 1 and pattern clustering. In G. Hinton, D. Touretzky and T. Sejnowsky, editors, Proceedings of the 1988 Connectionist Model Summer School, pages 174-185, San Mateo, C.A., 1989, Morgan Kaufmann.

41. J. Muruzábal and A. Muñoz. On the visualization of outliers via Self-Organizing Maps. journal of Computational and Graphical Statistics, 1996 (to appear).
42. A. Muñoz. Creating term associations using a hierarchical ART architecture. In C.v.d. Malsburg and W.v. Seelen, editors, International Conference on Artificial Neural Networks, Lecture Notes in Artificial Intelligence, vol. 1112, Bonn, Germany, 1996. Springer Verlag.
43. A. Muñoz and J. Muruzábal. Outlier detection via self-organizing maps. In Proceedings of the "New Techniques and Technologies for Statistics" Seminar, Bonn, Germany, November 1995.
44. H. Prade and C. Testemale. Fuzzy relational databases: Representational issues and reduction using similarity measures. Journal of the American Society for Information Science, 38(2):118-126, 1987.
45. V.V. Raghavan and S.K.M. Wong. A critical analysis of vector space model for information retrieval. Journal of the American Society for Information Science, 37(5):100-124, 1986.
46. G. Salton. Automatic text processing. Addison-Wesley, New York, 1989.
47. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Information Processing and Management, 24:513-523, 1988.
48. J.W. Sammon. A nonlinear mapping for data structure analysis. IEEE Transactions on Computers, 18(5):401-409, May 1969.
49. T. Saracevic and P. Kantor. A study of information seeking and retrieving. II. Users, questions and effectiveness. Journal of the American Society for Information Science, 39(3):177-196, 1988.
50. S.K.M. Wong and Y.Y. Yao. An information-theoretic measure of term specifity. Journal of the American Society for Information Science,43(1):54-61, 1992.
51. H.J. Zimmermann. Fuzzy set theory and its applications. Kluwer-Nijhoff Publishing, Dorddrecht, 2nd edition, 1990.
52. G.K. Zipf, Human behavior and the principle of least effort: An introduction to human ecology. Haffner, New York, 1972.